Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A computing device that provides hardware conversion of ~~flow~~ control flow to predicates associated with program instructions executable within said computing device, said computing device comprising:

predicate assignment means for detecting the beginning and the end of a branch domain of said program instructions, said predicate assignment means being invisible to instruction set architecture and thereby invisible to a user; and

predicate use means for realizing the beginning and the end of said branch domain at execution time, and for selectively enabling and disabling instructions within said branch domain, said predicate use means being invisible to instruction set architecture and thereby invisible to a user.

2. (New) The computing device according to claim 1 wherein said predicate assignment means includes a tracking buffer comprising dedicated register storage in order to make said predicate assignments.

3. (New) The computing device according to claim 1, wherein said predicate assignment means is operative to assign a canceling predicate to said branch domain in order to delineate said branch domain so that effects of its corresponding branch are nullified.

4. (New) The computing device according to claim 3, wherein said predicate use means further includes dedicated registers for each instruction in order to effect arbitrary control flow, including least disjoint, nested, overlapped and combinations of said branch domains.

5. (New) A method for providing hardware conversion of control flow to predicates in order to enable program instructions to be executable within a computing device, said method comprising:

detecting the beginning and the end of a branch domain of selected said program instructions in a manner that is invisible to instruction set architecture and thereby is invisible to a user;

generating from each said branch domain a predicate;

associating said predicate with at least one program instruction; and thereafter

realizing the beginning and the end of said branch domain at execution time thereby selectively enabling and disabling instructions within said branch domain.

6. (New) The method according to claim 5 wherein said detecting step includes using a tracking buffer to make said predicate assignments.

7. (New) The method according to claim 5 wherein said predicate generating step assigns a canceling predicate to said branch domain in order to delineate said branch domain so that effects of its corresponding branch are nullified.

8. (New) The method according to claim 7, wherein said predicate generating further includes using dedicated registers for each instruction in order to effect arbitrary control flow, including least disjoint, nested, overlapped and combinations of said branch domains.